Temperature Indicator for the PC

Useful and decorative!

Design by A. Köhler

Alongside its purely functional uses the PC also has a position as a status symbol. At computer nerd get-togethers, it is *de rigeur* to have some novelty to show off. And if it also serves some useful purpose, so much the better.

Enclosures with windows and internal illumination have already started to become available, showing off the innards of a PC. There is even a special word — *modding* — for this type of PC (re)construction. As well as various modifications whose benefits can be argued over at considerable length, there are those which have a definite useful purpose.

For example, the temperature inside a PC's enclosure is an important quantity. Depending on the processor, the fans and the ambient conditions, the temperature can rise to the point where trouble-free operation can no longer be guaranteed. There are special displays available, often in conjunction with fan systems, that offer a temperature readout. Unfortunately these displays occupy a full 3.5 inch or 5.25 inch drive bay in the PC. But remember that PCs have a red, yellow or green LED that serves as a power indicator. Why not use this simple indicator to offer extra functions, such as showing the temperature inside the enclosure?

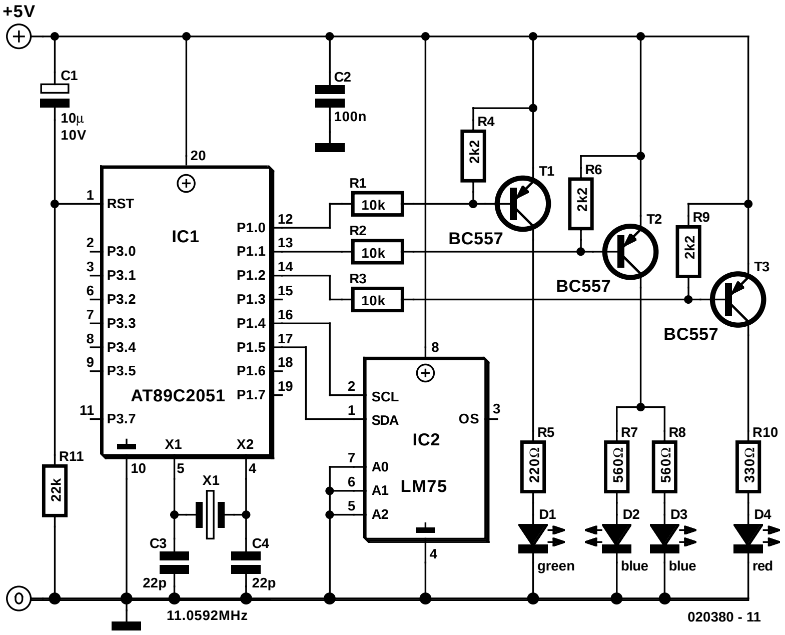

Figure 1. The simple circuit diagram of the temperature indicator.

Before we start

Before we can build a temperature indicator we must determine the range of temperatures over which it is to operate. Intel and AMD specify their processors for a temperature of 40 °C to 45 °C inside the case. Without expensive specialised cooling it is not really possible to effectively remove the heat generated by the processor at case temperatures above 70 °C. The maximum junction temperature of 125 °C is in any case the ultimate limit. An office PC will usually be used in a more-or-less air-conditioned environment (the *Elektor Electronics* offices being a particularly notable exception). And hardly any user will want to operate their PC at temperatures below 15 °C, so the PC's components are rarely below this temperature.

It is therefore not at all unreasonable to set the measurement range to +10 °C to +70 °C. Using these values we can use the top surfaces of hard disks and heatsinks as measurement points. The temperature sensor used should be a readily available and inexpensive type, such as the LM75 (or DS75). Reading the temperature out can be done using a simple microcontroller such as the AT89C2051, which will allow us to transfer the temperature value over an I^{2}C bus. We can use a three-colour LED as an indicator, replacing the one-colour power indicator LED usually provided. The multicolour LED is constructed from four LED chips in three colours: one red, one green, and, because of their lower light output, two blue chips. In order to give the impression of a single composite colour, a diffused package is used. If the individual LED chips can still be made out, adding an extra sheet of diffuser will result in the desired effect. The impression of different colours is achieved by driving the individual LED chips using pulse-width modulated signals from the microcontroller.

The circuit

Figure 1 shows the complete circuit, using an 89C2051 microcontroller from Atmel. This device's baby brother, the 89C1051 (which offers 1 kbyte of memory) can also be used. The reset circuit on pin 1 is not critical: the capacitor value can be anything from 1 µF to 10 µF, the resistor anything from 10 kΩ to 50 kΩ. Likewise, the oscillator circuit between pins 4 and 5 is not critical: crystals from 7.3728 MHz to 14.318 MHz have been tested. The three-colour LED is connected via a PNP transistor driver. A 5 mm LED in a diffused package is used: suitable devices are available from, for example, Reichelt Electronics (www.reichelt.de) and Conrad Electronics (order code 185353, www.int.conradcom.de). The LED is manufactured by Kingbright under part code LF59EMBGMBW. Note that many devices advertised as 'three-colour LEDs' are not suitable for this application. The Kingbright device you need has six terminals, see also the datasheet which may be found on the Conrad webserver.

Each BC557 transistor has a series base resistor and a pull-up resistor to ensure that the transistor is turned fully off. Because of the different forward voltages and different light intensities of the LEDs, different values of current-limiting resistor are required. These resistor values can be adjusted, since the LED has reasonably wide tolerances.

Port 1.0 controls the green LED, port 1.1 the blue LED and port 1.2 the red LED. The LED current should be around 20 mA. Communication with the temperature sensor is over ports 1.4 (the SCL clock signal) and 1.5 (the SDA data signal). The LM75CIM5 temperature sensor (the 5 V version) has its write address set to 90H and its read address set to 91H by the levels on A2 to A0.

Software

The most interesting part of this project is the software. The software is available as source and compiled on the Internet from the *Elektor Electronics* website (which is free) and on diskette **020380-11** (for which there is a nominal charge). If you cannot, or do not wish to, program the device yourself, a ready-programmed microcontroller is also available (order number **020380-41**).

The details of the program can be seen from the listing. The program uses pulse-width modulation, controlled by a program loop rather than using the internal timers. The program begins by initialising the port outputs. A register is allocated to each colour to store the time for which the corresponding LED is illuminated. Next the program selects the temperature register of the LM75 and reads out the temperature value stored there. Only the four most significant bits are used, the four least significant bits of the temperature reading being ignored. Since the full temperature range of the device is not used, the temperature scale is shifted to start at 10 °C by subtracting 0AH from the value read. This value is to be converted into a 'colour value'. Each colour value consists of three time constants which determine for how long the corresponding primary colour LED is illuminated. The time constants are obtained from a table indexed by the temperature. Routine ZAU causes the colour value to be displayed.

The data pointer is used to point into the table containing the illumination time values. Each temperature value uses three bytes of table space. The first byte gives the time for which the green LED is illuminated. The green LED is turned on and a delay loop is entered. The same happens for the blue and red LEDs. The primary colour LEDs are driven inside a loop which is executed 255 times, so that the eye cannot distinguish the individual colours. Instead, the impression of a particular colour and intensity is given, depending on the time constants and hence on the temperature.

A particular point to note is that a value of 00H for a time constant corresponds to the maximum possible illumination period, since the first time the DJNZ instruction is executed in the loop an overflow (or rather, a borrow) occurs. With a value of 01H the illumination period is so brief that it is practically invisible to the eye. Delay routine ZEIT is used to extend the (otherwise very short) illumination period. If the crystal frequency is changed, it may be necessary to adjust this delay somewhat. The values given have been tested with a 11.0592 MHz crystal. The colours themselves can be changed by altering the values in the table.

The example program given indicates low temperatures by a blue colour. As the temperature rises, so does the proportion of green in the colour. At 35 °C the green LED alone is lit. If the temperature climbs further, red is added into the colour. Over 50 °C only the red LED lights.

Since the maximum offset available is 255 (an 8 bit register is used for the temperature), the table can only be extended to about 70 °C, but above this temperature an office PC is unlikely to operate correctly in any case. Communication with the temperature sensor uses the I^{2}C bus routines START, STOP, READ and WRITE in their minimal configuration. Error handling is not provided as transmission errors on the I^{2}C bus will not have serious consequences. With the source listing to hand there should be no difficulty in making modifications to the display.

(020380-1)

Free Download

source code file (in German) and hex file for programming the microcontroller. File number: 020380-11.zip.
www.elektor-electronics.co.uk/dl/dl.htm
(select month of publication).